

# Asymmetric quantum error correcting codes

Lev Ioffe<sup>1</sup> and Marc Mézard<sup>2</sup>

<sup>1</sup>*Center for Materials Theory, Department of Physics and Astronomy,  
Rutgers University 136 Frelinghuysen Rd, Piscataway NJ 08854 USA*

<sup>2</sup>*CNRS; Univ. Paris-Sud, UMR 8626, LPTMS, Orsay Cedex, F-91405 FRANCE*  
(Dated: February 1, 2008)

The noise in physical qubits is fundamentally asymmetric: in most devices, phase errors are much more probable than bit flips. We propose a quantum error correcting code which takes advantage of this asymmetry and shows good performance at a relatively small cost in redundancy, requiring less than a doubling of the number of physical qubits for error correction.

PACS numbers: 03.67

*Introduction.* The quest of a quantum computer has stimulated a lot of interesting developments in recent years. However, despite a remarkable progress, all physical devices realized so far do not allow to build even a very small computer. One crucial aspect is the noise control. Quantum computing faces two antagonist constraints: one should be able to manipulate and address the results of a computation, and at the same time one must keep the noise level low. While some hardware architecture may help to achieve this compromise, it is clear now that there will never exist a quantum computer without efficient quantum error correction (QEC).

The basic principles of QEC have been written down in [1, 2, 3, 4], and a number of QEC codes have been developed since then [5, 6]. However, most of them require in practice a high level of redundancy (in coding language, a low rate): the number of physical qubits needed to effectively protect one logical qubit is large. Generally one expects that a higher rate might be achievable with good codes when the length of the information block is large: in this limit, the uncorrected errors correspond to a correlated flip of a large number of physical bits, the probability of which gets exponentially small. The classical coding theory shows the existence of codes that become ideal (saturate the Shannon limit) when the size of the block tends to infinity. Furthermore, recent progress on so-called low density parity check (LDPC) codes has revived older ideas by Gallager [7] that produce efficient algorithms for a fast decoding of codes with performance close to Shannon's limit [8, 9, 10]. The generalization of these classical schemes for quantum error correction is made very difficult by the requirement that a quantum scheme should correct two types of errors: bit flips as well as phase errors. Another important constraint is the difficulty to perform operations concurrently on the same bit: an efficient error correction scheme should involve a relatively small ( $o(N)$ ) read-out operations on each bit. Here we propose a new family of QEC

codes which work at relatively low redundancy (typically  $\leq 2 - 3$  physical qubits for one logical qubit), can correct many mistakes and allow parallelization. So far, the main attempt at finding such codes is the work [11]. It uses so-called self-dual codes which are tailored to deal with a noise which is symmetric in all channels. We argue that in the physical devices conceived so far, the noise is typically asymmetric (a phase error is much more probable than a bit flip), and one can exploit this asymmetry to develop more efficient QEC codes. The construction that we propose here makes use of two standard classical codes which are among the most efficient ones: it handles the relatively rare bit errors through a Bose Chaudhuri Hocquenghem (BCH) code [12] and the more frequent phase errors through a LDPC code, with performances close to those of the most powerful random LDPC codes[10].

*Physical noise* The level of the noise in a single physical bit is conveniently characterized by the relaxation time,  $T_1$ , and dephasing time  $T_2$ , the two parameters that enter Bloch equation for a single bit (spin) dynamics. Because the relaxation always implies dephasing, the dephasing rate  $1/T_2$  has a contribution from the relaxation processes and a pure dephasing:  $1/T_2 = 1/(2T_1) + \Gamma_\phi$ . Generally, there are many ways to control the relaxation rate: first, the relaxation between two states with energy difference  $\Delta E$  requires a transfer of energy to the environment, the amplitude of which becomes smaller as  $\Delta E \rightarrow 0$ . Furthermore, in many physical implementations these two states are separated by a large barrier that makes transitions between them rare. The situation is completely different with the dephasing rate  $\Gamma_\phi$  which is physically due to the fluctuations of  $\Delta E$  with time. All low frequency processes contributing to the  $\Delta E(t)$  dependence result in the decrease of  $\langle \exp(-i \int \Delta E(t) dt) \rangle$  correlator, i.e. lead to the dephasing. In this respect, a particularly damaging effect comes from omnipresent  $1/f$  noise. Thus, it is not surprising that in almost all devices

studied so far, the relaxation rate can be made much slower than the dephasing: in a typical NMR device  $T_1 \sim 10 - 100s$  while  $T_2 \sim 1s$  [16], in superconducting phase qubits  $T_1 \sim 10\mu s$  while  $T_2 \sim 100ns$  [17], in superconducting charge qubit  $T_1 \sim 100ns$  while  $T_2 \sim 1ns$  [18] and finally for spin dots  $T_1 \sim 1\mu s$  while  $T_2 \sim 10ns$  [19, 20].

In the following we shall therefore assume that in physical qubits the noise is strongly asymmetric. Specifically, we study a noise channel defined as follows. Noise acts independently on each bit. It induces a bit flip with probability  $p_x$ , and independently it induces a phase flip with probability  $p_z$ . The original state of the system,  $|\psi_0\rangle$ , is thus changed to  $|\psi\rangle = \prod_i [(\sigma_z^i)^{m_i} (\sigma_x^i)^{n_i}] |\psi_0\rangle$  with probability  $p_z^{\sum_i n_i} (1 - p_z)^{N - \sum_i n_i} p_x^{\sum_i m_i} (1 - p_x)^{N - \sum_i m_i}$ , where  $m_i, n_i \in \{0, 1\}$ . The channel acts on bit  $i$  by applying an operator  $U_i \in \{\mathcal{I}, \sigma_x^i, \sigma_z^i, \sigma_x^i \sigma_z^i\}$ .

*CSS codes.* Our family of codes is of the CSS type [3, 4]. It consists of two independent encoding/decoding devices dealing separately with bit and phase flips, for a string of  $N$  physical qubits. It uses  $M_z$  'z-checks' and  $M_x$  'x-checks'. The  $a$ -th z-check is defined by a set  $V(a) \in \{1, \dots, N\}$  and by the operator  $C_a^z = \prod_{i \in V(a)} \sigma_z^i$ . Similarly, the  $a$ -th x-check is defined by a set  $W(a) \in \{1, \dots, N\}$  and by the operator  $C_a^x = \prod_{i \in W(a)} \sigma_x^i$ .

By construction, the z-checks and x-checks all commute with each other, and the original state  $|\psi_0\rangle$  is an eigenstate of all the operators  $C_a^z, C_{a'}^x$  with eigenvalue 1. As  $U_i$  either commutes or anticommutes with these check operators, the noise-perturbed state  $|\psi\rangle$  is an eigenstate of the operators  $C_a^z, C_{a'}^x$ . The decoding operation consists of three steps: (i) measure the eigenvalues of the check operators, ii) infer from these eigenvalues what was the corrupting operator, (iii) apply the correction operator. In more detail:

Step (i): The  $a$ -th z-syndrom is defined as the number  $u_a \in \{0, 1\}$  such that  $C_a^z |\psi\rangle = (1 - 2u_a) |\psi\rangle$ . Similarly, the  $a$ -th x-syndrom is defined as the number  $v_a \in \{0, 1\}$  such that  $C_a^x |\psi\rangle = (1 - 2v_a) |\psi\rangle$ .

Step (ii): From the z-syndroms  $\{u_a\}$ ,  $a \in \{1, \dots, M_z\}$ , we compute  $N$  numbers  $\{m'_1, \dots, m'_N\}$  such that, for each  $a \in \{1, \dots, M_z\}$ :  $\sum_{i \in V(a)} m'_i = u_a \pmod{2}$ , with the smallest possible number of  $m'$ 's equal to 1. From the x-syndroms  $\{v_a\}$ ,  $a \in \{1, \dots, M_x\}$ , we compute  $N$  numbers  $\{n'_1, \dots, n'_N\}$  such that, for each  $a \in \{1, \dots, M_x\}$ :  $\sum_{i \in W(a)} n'_i = v_a \pmod{2}$ , with the smallest possible number of  $n'$ 's equal to 1.

Step (iii): generate  $|\psi'\rangle = \prod_{i=1}^N [(\sigma_x^i)^{n'_i} (\sigma_z^i)^{m'_i}] |\psi\rangle$ . If the error correction is

successful, one should find  $|\psi'\rangle = |\psi_0\rangle$ .

A CSS code is thus characterized by the sets  $V(a)$  and  $W(a)$  defining the checks. In building such a code, one must ensure that all check operators commute. This imposes that  $\forall a \in \{1, \dots, M_z\}$ ,  $\forall a' \in \{1, \dots, M_x\}$ , the cardinal of  $|V(a) \cup W(a')|$  be even. It is useful to define the parity check matrices of the two codes. The matrix  $H^z$  is a  $M_z \times N$  matrix with entries in  $\{0, 1\}$ , defined by  $H_{ai}^z = 1$  if and only if  $i \in V(a)$ . Similarly,  $H^x$  is the  $M_x \times N$  matrix defined by  $H_{ai}^x = 1$  if and only if  $i \in W(a)$ . The commutativity condition is satisfied when  $H^z (H^x)^T = 0$  (using Boolean algebra, i.e. mod(2) additions). The z-codewords are strings of  $N$  bits  $x_i \in \{0, 1\}$  such that,  $\forall a$ ,  $\sum_i H_{ai}^z x_i = 0 \pmod{2}$ . Any x-check  $a$  defines a z-codeword through  $x_i = 1$  if  $i \in W(a)$ , and  $x_i = 0$  otherwise. Similarly, z-checks define x-codewords. Most of the research on QEC so far has focused on the design of relatively small codes with good distance properties. If for instance all pairs of x-codewords are at a Hamming distance  $\geq 2d+1$ , the code will correct any set of  $\leq d$  flip errors. While this suggest to build codes which maximize the smallest distance between codewords, this strategy is not necessarily optimal when dealing with large blocklength ( $N \gg 1$ ). Instead, what is practically required is that the probability of an error is small and it turns out that the best classical codes often have (very rare) pairs of codewords which are pretty close to each other [10]. We shall use this approach to construct our x-checks.

*z-checks: BCH code* Our z-code is an efficient classical construction, a binary primitive BCH code (see ref. [13] for an extended presentation). The code depends on two parameter  $m, t$ . The first one determines the Galois field  $GF(2^m)$  which is used, and the number  $t$  is equal to the number of errors (phase flips) that the code can correct. The number of variables (and therefore the number of qubits) is given by  $N = 2^m - 1$ . If  $\alpha$  is a primitive element of the field  $GF(2^m)$ , the powers  $\alpha^r$ ,  $r \in \{1, \dots, N\}$  are  $N$  distinct elements of the field, building a cyclic group under multiplication. At the same time,  $GF(2^m)$  is a vector space of dimension  $m$  over  $GF(2)$ : every element  $\alpha^r$  can be decomposed as  $\alpha^r = \sum_{p=0}^{m-1} \gamma_{rp} \alpha^p$ , where the coefficients  $\gamma$  are in  $\{0, 1\}$ . The check matrix  $H$  of the code is defined as

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{N-1} \\ 1 & (\alpha^3) & (\alpha^3)^2 & \dots & (\alpha^3)^{N-1} \\ 1 & (\alpha^5) & (\alpha^5)^2 & \dots & (\alpha^5)^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & (\alpha^{2t-1}) & (\alpha^{2t-1})^2 & \dots & (\alpha^{2t-1})^{N-1} \end{bmatrix} \quad (1)$$

This matrix can be seen either as a  $t \times N$  matrix with elements in  $GF(2^m)$ , but another interpreta-

tion is also useful. If we write each element  $\alpha^r$  of  $H$  as the  $m$  component vector  $\begin{pmatrix} \gamma_{r0} \\ \dots \\ \gamma_{r(m-1)} \end{pmatrix}$ , we obtain the  $tm \times N$  parity check matrix  $H^z$  with entries in  $GF(2) = \{0, 1\}$ . Therefore  $M_z = tm$ . BCH decoding relies on algebraic properties which are most easily written in terms of polynomials. Here we shall just present the basic result in the case  $t = 2$ . If two of the  $N$  bits are flipped by noise, and these indices correspond to the elements of  $GF(2^m)$  called  $\beta_1, \beta_2$ , the check matrix  $H$ , applied to the error vector, gives two syndromes  $\zeta_1 = \beta_1 + \beta_2$  and  $\zeta_3 = \beta_1^3 + \beta_2^3$ . Decoding consists in finding  $\beta_1, \beta_2$  given  $\zeta_1, \zeta_2$ . It is easily seen that this system has a unique solution in  $GF(2^m)$  (up to the permutation of  $\beta_1$  and  $\beta_2$ ): the code with  $t = 2$  corrects exactly any set of  $\leq 2$  errors. The same construction works for arbitrary  $t$ , and good decoding algorithms exist: the code corrects any set of  $\leq t$  errors. In practice we have used the Berlekamp algorithm [13], adapting some software available from [15].

*Generation of the  $x$  checks: LDPC code.* Some BCH codes are self-dual; in such a case one gets a quantum code using  $H^x = H^z$  [14]. But in order to get a much better performance (for large  $N$ ) on the  $x$ -channel, we prefer to use a code as close as possible to the random LDPC codes. The commutation of the  $x$  and  $z$  checks is obtained by the following procedure. Given a BCH code with parameters  $m, t$ , we can generate a  $x$ -check  $a$  with any degree  $n \geq 2t + 1$  using a variant of its standard decoding algorithm. The first  $n - t$  elements of  $W(a)$  are chosen as a random subset of  $\{1, \dots, N\}$  with distinct elements, taken uniformly among all such subsets. Let us call  $\beta_1, \dots, \beta_{n-t}$  the corresponding elements of  $GF(2^m)$ . We look for the remaining  $t$  elements which are solutions of the decoding equations

$$\forall s \in \{1, \dots, t\} : \sum_{r=1}^t (\beta_{n-t+r})^{2s-1} = - \sum_{r=1}^{n-t} (\beta_r)^{2s-1}.$$

Provided that the solution of these equations exists (which happens with probability  $1/t!$ ) the elements  $\beta_{n-t+1} \dots \beta_n$  can be found using any standard BCH decoding algorithm, like Berlekamp's one. The indices corresponding to the elements  $\beta_1, \dots, \beta_{n-t} \dots \beta_n$  form the subset  $W(a)$  defining the  $a$ -th  $x$ -check. As  $\beta_1, \dots, \beta_n$  is a codeword of the BCH code, the commutativity condition is satisfied.

Clearly, the indices in  $V(a)$  do not form a random subset of size  $n$ . However, if the map used in generating  $\beta_{n-t+1} \dots \beta_n$  from  $\beta_1, \dots, \beta_{n-t}$  is chaotic enough (we shall refer to this hypothesis as the 'chaos hypothesis' in the following), one can hope to generate a set of  $x$ -checks with performances close

to the ones of classical random LDPC codes. This is what we have found numerically. In practice, for a given value of  $t$ , we generate a large enough pool of parity checks, all having degree  $n = 2t + 1$ . From this pool, we select a number  $M_z$  of checks, in such a way that the degrees of the variables in the corresponding factor graph has a narrow distribution. This is done by the following inductive procedure. At each step we order the remaining (unused) set of checks by their 'quality' which is defined as the number of minimal degree variables that would be affected by addition of this check. We then add one (randomly chosen) check of the highest 'quality' and repeat the procedure.

The practical decoding of our LDPC code uses the standard 'belief propagation' (BP) algorithm [8, 9], a message passing algorithm which is equivalent to an iterative solution of Bethe equations.

*Performance.* An important parameter of the code is its degree of redundancy. We have checked that the various checks are generically linearly independent, so the  $z$ -rate (resp.  $x$  rate) is obtained as  $R_z = 1 - \frac{M_z}{N}$  (resp  $R_x = 1 - \frac{M_x}{N}$ ) and the quantum rate of the code is  $R = 1 - \frac{M_x + M_z}{N}$ .

The error correction ability depends on the channel. In the  $z$ -channel (bit flip errors), by construction, the BCH code is able to decode up to  $t$  errors. Therefore the probability of error in decoding this channel is

$$P_{\text{err}}^z = \sum_{j=t+1}^N \binom{N}{j} p_z^j (1 - p_z)^{N-j}, \quad (3)$$

which is well approximated, for the small values of  $p_z$  which interest us here, by  $1 - e^{-N p_z} \sum_{j=0}^t (N p_z)^j / j!$ .

Let us now turn to the  $x$ -channel. The performance of BP decoding for random LDPC codes can be studied analytically in the limit of large block-length [10]. Within the chaos hypothesis, one could thus derive the threshold for zero error decoding in the large  $N$  limit. However in practice we are interested in not-too-large values of  $N$ . We have thus tested numerically the BP decoding of our  $x$ -code.

The simulation is run as follows. We fix an 'acceptable' value of the block error  $P_{\text{block}}$  for decoding  $N$  bits, both in the  $x$  and in the  $z$ -channel, in practice  $P_{\text{block}} = 10^{-4}$ . For given values of  $N$  (or  $m$ ) and  $t$ , eq.(3) gives the noise level  $p_z$  that can be corrected in the  $z$ -channel, and the channel asymmetry gives the ratio  $p_z/p_x$ . We then test various  $x$ -codes, varying  $M_x$  until the block error in the  $x$ -channel is less than  $P_{\text{block}}$ . Results are summarized in the following table, which studies asymmetries  $p_z/p_x = 0.01, 0.1$ .

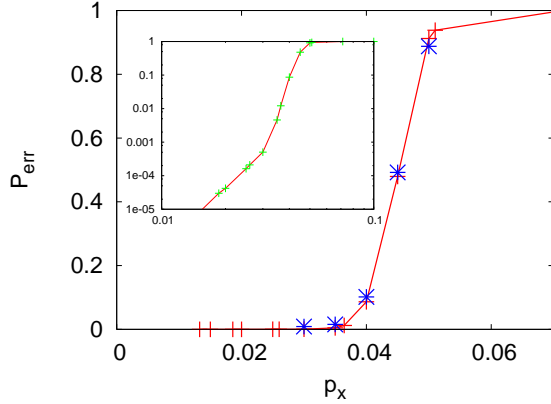


FIG. 1: +: block error in the  $x$ -channel,  $P_{\text{err}}^x$ , versus the phase error probability  $p_x$ , for the code with  $m = 12$ ,  $t = 4$ ,  $M_x = 1378$ . The line is a guide to the eye. Also shown is the same curve for a random LDPC code ( $\times$ ). The inset gives the same data in a log-log plot

$m$	$t$	$p_z$	$M_z$	$Q_z$	$p_x$	$M_x$	$Q_x$	$Q$
10	2	$8.40 \cdot 10^{-5}$	20	.980	$8.4 \cdot 10^{-3}$	563	.45	.43
10	3	$2.26 \cdot 10^{-4}$	30	.971	$2.26 \cdot 10^{-2}$	460	.55	.52
10	4	$4.34 \cdot 10^{-4}$	40	.961	$4.34 \cdot 10^{-2}$	530	.48	.44
10	3	$2.26 \cdot 10^{-4}$	30	.971	$2.26 \cdot 10^{-3}$	460	.55	.52
10	4	$4.34 \cdot 10^{-4}$	40	.961	$4.34 \cdot 10^{-3}$	344	.66	.62
10	5	$6.98 \cdot 10^{-4}$	50	.951	$6.98 \cdot 10^{-3}$	271	.73	.69
10	6	$1.01 \cdot 10^{-3}$	60	.941	$1.01 \cdot 10^{-2}$	285	.72	.66
12	3	$5.66 \cdot 10^{-5}$	36	.991	$5.66 \cdot 10^{-3}$	1577	.61	.61
12	4	$1.08 \cdot 10^{-4}$	48	.988	$1.08 \cdot 10^{-2}$	1378	.66	.65
12	5	$1.74 \cdot 10^{-4}$	60	.985	$1.74 \cdot 10^{-2}$	1189	.71	.69
12	6	$2.52 \cdot 10^{-4}$	72	.982	$2.52 \cdot 10^{-2}$	1191	.71	.69

We see that large enough codes provide a good performance. For instance,  $m = 12, t = 6$ , code with  $N = 4095$  qubits is able to correct a noise level of  $p_z = 2.5 \cdot 10^{-4}$  in the  $z$  channel and  $p_x = 2.5 \cdot 10^{-2}$  in the  $x$  channel with block error probability smaller than  $10^{-4}$ . Notice that for these values of  $p_z, p_x$ , the probability of a block error *without* any error correction would be  $1 - (1 - p_{z,x})^N$ , giving .63 for the  $z$ -channel and 1 for the  $x$ -channel. Figure (1) gives the block error in the  $x$ -channel,  $P_{\text{err}}^x$ , versus the phase error probability  $p_x$ , for one given code.

**Conclusions.** We have provided an explicit construction of quantum codes with rates  $Q \sim 0.5$  that are able to correct a few errors in one channel (bit flips) and have close to optimal performance in another (phase errors), together with efficient decoding procedures. One important aspect of these codes is the fact that the number of operations to be done on one given bit is much smaller than  $N$ . In the  $z$ -channel this is due to the fact that we use a small value of  $t$ , in the  $x$ -channel it is due to the intrinsic

low density of the code. We believe that these codes might be quite useful for the realistic physical implementation of quantum memory. We have not investigated the possibility of using them for fault tolerant quantum computation, this is the subject of the future research.

**Acknowledgments.** We thank J.S. Yedidia for interesting discussions. This work has been supported in part by the EC grants 'Stipco', HPRN-CT-2002-00319, 'Evergrow', IP 1935 in the FET-IST programme and NSF DMR 0210575. LI thanks LPTMS for the hospitality that made this work possible.

- 
- [1] P.W. Shor, Phys.Rev. A **52** (1995) R2493
  - [2] A.M. Steane, Phys. Rev. Lett. **77** (1996) 793
  - [3] A. R. Calderbank and P.W. Shor, Phys. Rev. A **54** (1996) 1098
  - [4] A.M. Steane, Proc. Roy. Soc. Lond. A **452** (1996) 2551
  - [5] See e.g: A.R. Calderbank, E.M. Rains, P.W. Shor and N.J.A Sloane, IEEE Trans. Inform. Theory, **44** (1998) 1369 and references therein
  - [6] A. Yu. Kitaev, quant-ph/9707021 (1997), Ann. Phys. **303**, 2 (2003); E. Dennis, A. Kitaev, A. Landahl and J. Preskill, J. Math. Phys **43** (2002) 4452
  - [7] R.G. Gallager "Low-Density Parity-Check Codes", MIT Press, Cambridge, MA (1963)
  - [8] D.J.C. MacKay "Information theory, Inference and Learning Algorithms", Cambridge Univ. Press 2003
  - [9] T. Richardson and R. Urbanke, "Modern Coding Theory", in preparation at <http://lthcwww.epfl.ch/mct/index.php>
  - [10] *Special Issue on Codes, Graphs and Iterative Algorithms*, IEEE Trans. Info. Theory **47** no.2 (2001).
  - [11] D.J.C. MacKay, G. Mitchison and P.L. McFadden, IEEE Trans. Inf. Theory **50** (2004) 2315
  - [12] R.C. Bose and C.R. Ray-Chaudhuri, Inform.Control **3** (1960)68; A. Hocquenghem, Chiffres **2** (1959) 147.
  - [13] S. Lin and D.J. Costello, "Error control coding", Pearson Prentice Hall, 2004.
  - [14] A.M. Steane, IEEE Trans. Inf. Theory **45** (1999) 2492
  - [15] R. Morelos-Zaragoza, <http://www.eccpage.com/>
  - [16] L.M.K. Vandersypen, M. Steffen, G. Breyta, C.S. Yannoni, M.H. Sherwood and I.L. Chuang, Science **414**, 883 (2001).
  - [17] P. Bertet, I. Chiorescu, G. Burkard, K. Semba, C.J.P.M. Harmans, D.P. DiVincenzo and J.E. Mooij Phys. Rev. Lett. **95**, 257002 (2005).
  - [18] O. Astafiev, Yu. A. Pashkin, Y. Nakamura, T. Yamamoto and J. S. Tsai, Phys. Rev. Lett. **93**, 267007 (2004).
  - [19] J.M. Elzerman, R. Hanson, L.H. Willems van Beveren, B. Witkamp, L.M.K. Vandersypen and L.P. Kouwenhoven, Nature **430**, 431 (2004)
  - [20] Y. Kato, R.C. Myers, A.C. Gossard and D.D. Awschalom, Nature **427**, 50 (2004)